Key Management Schemes for Secure Communication in Heterogeneous Sensor Networks

A.S.Poornima[1], B.B.Amberker[2]
[1] Dept. of Computer Science and Engg, Siddaganga Institute of Technology,
Tumkur, Karnataka, India.
Email: aspoornima@sit.ac.in
[2] Dept. of Computer Science and Engg, National Institute of Technology,
Warangal, Andhra Pradesh, India.
Email: bba@nitw.ac.in

***Abstract*—Hierarchical Sensor Network organization is widely used to achieve energy efficiency in Wireless Sensor Networks(WSN). To achieve security in hierarchical WSN, it is important to be able to encrypt the messages sent between sensor nodes and its cluster head. The key management task is challenging due to resource constrained nature of WSN. In this paper we are proposing two key management schemes for hierarchical networks which handles various events like node addition, node compromise and key refresh at regular intervals. The Tree-Based Scheme ensures in-network processing by maintaining some additional intermediate keys. Whereas the CRT-Based Scheme performs the key management with minimum communication and storage at each node.**

***Index Terms*—Hierarchical Sensor Networks, Chinese Remainder Theorem, Cluster Key, Cluster head, Sensor Node.**

I. INTRODUCTION

Wireless Sensor Networks (WSN) are composed of small autonomous devices, or sensor nodes, that are networked together. Sensor networks can facilitate large-scale, real-time data processing in complex environments. Their applications involve protecting and monitoring critical military, environmental, safety-critical or domestic infrastructures and resources. Wireless communication employed by the WSN facilitates eavesdropping and packet injection by an adversary. This factor demand security for sensor network to ensure operation safety, secrecy of sensitive data and privacy for people in sensor environment.

The key management schemes discussed in [1,2,3,6,7, 9,11] consider homogeneous sensor networks, where all sensor nodes have identical capabilities in terms of communication, computation and storage. Large scale homogeneous networks suffer from high costs of communication, computation and storage requirements. Hence Hierarchical Sensor Networks (HSNs)(also called as Heterogeneous Sensor Networks) are preferred as they provide better performance and security solutions. In WSN's hierarchical clustering provides scalability, self-organization and energy efficient data dissemination. In hierarchical networks, there is a hierarchy of nodes in terms of resources and functions. The most powerful node is the Base Station (BS). BS is a powerful data processing and storage unit which collects sensor readings, perform costly operations and manage the network. It interfaces the network to outside world. Transmission power of BS is usually enough to reach all nodes. The next level of sensors are called *group heads* or *Cluster Heads* (we call these nodes as CH-sensors). These nodes have better resources compared to the sensor nodes which form the lowest level of this model. Cluster heads are responsible for intermediate data processing, data aggregation e.g. collect and process the readings of other nodes in the cluster and send a single reading to base station. The BS in turn performs computation on readings from multiple cluster heads. The sensor nodes i.e., nodes with least resources and used for sensing a particular data (called as SN-sensors) form the majority of the network. They provide the readings for the parameters being sensed.

Hierarchical Sensor Networks(HSN) are considered in [8,4,10,5]. In the scheme [8] proposed by Sajid et.al. key management based on key pre distribution is discussed. Routing driven key management scheme is discussed in [4], the scheme is based on Elliptic Curve Cryptography. The scheme [10] focuses on achieving higher key connectivity and system performance using the combination of nodes with higher capability and nodes with lower capability in terms of computation, communication and storage. In [5] algorithms are discussed to improve the degree of sensing coverage using heterogeneous sensor networks.

In this paper we are proposing key management schemes for Heterogeneous Sensor Networks. The first scheme is called as *tree based scheme* which is based on the scheme in [13]. The second scheme is based on Chinese Remainder theorem which is proposed for wired networks [14] which is called as *CRT-based scheme* in this paper. Here, hierarchical architecture of sensor networks is considered, where data is routed from sensor nodes to the base station through cluster head. Base station interfaces sensor network to the outside network. Sensor nodes are assumed to be immobile, these nodes organize themselves into clusters. The size of the cluster we are assuming here is a small group of sensor nodes. A cluster head is chosen from each cluster to handle the communication between the cluster nodes and the base station. In the key management scheme discussed in [8] node revocation is not considered in detail. This scheme

DOI: 01.ijns.01.01.05

discusses about the percentage of links that are compromised when a node is compromised, but how these compromised links are reconfigured and what is the effort involved to reconfigure the compromised links in not discussed. The proposed schemes present how actually the keys are changed (*rekey operation*: is nothing but changing the keys that are known to compromised node and distributing them securely to existing nodes) in order to reconfigure the compromised links when a node is compromised.

The proposed schemes are analyzed in detail by considering various performance metrics like storage, communication and computation. The analysis shows that *Tree-Based scheme* achieves *rekey* operation by performing $\log_m$ n communication with additional storage, whereas [8] achieves the same goal using 2n communication. The *CRT-Based scheme* achieves *rekey* operation by performing one modulus and one EX-OR operation and no additional communication cost is incurred.

The paper is organized as follows : In Section II we explain notations, security goals and the threat model. Section III explains in detail the *Tree-Based Scheme*. In Section IV we explain the *CRT-Based Scheme*. Section V presents the performance analysis of the proposed schemes and finally we conclude in Section VI.

II. System Model

In this section we discuss about assumptions and notations, security goals and threat model used in this paper to construct the key management schemes.

A. Notations

Following are some of the notations used in this paper :

BS	→	Base Station
CH	→	Cluster Head
S	→	Set of all sensor nodes in a cluster
CCHK	→	Common Cluster Head Key
n	→	Number of nodes in a cluster
s_i	→	i^{th} Sensor node
CK	→	Cluster Key
k_i	→	Private key of the ith sensor node
k_{i-j}	→	Key k shared between the users from I to j
$\{x\}_y$	→	Encryption of x using key y
K	→	Set of pairwise relatively prime numbers

B. Security Goals

The main security goal considered in this paper is confidentiality : only the authorized nodes should be able to read the messages transmitted between the nodes. The confidentiality requirements that we are achieving in the *Tree-Based Scheme* and *CRT- Based Scheme* are :

1. **Non-group confidentiality :** Nodes that are not in the cluster should not be able to access any key that can be used to decrypt the message sent to the legitimate nodes.
2. **Forward confidentiality :** When a node is compromised, the scheme should ensure that the compromised node does not have access to any key used to encrypt the future messages.
3. **Backward confidentiality :** When a new node is added to the cluster, the scheme should ensure that the node does not have access to any key such that it can decrypt the previous messages.

C. Threat Model

The type of attacker we are considering in this paper are of two types. First type of attacker is an *outside attacker* who is able to eavesdrop on the communications. Second type of attacker is *inside attacker* a compromised node which is able to get all the secrets.

III. Description of the Tree - based scheme

Sensors within a cluster are organized as *m*-ary [13] balanced tree with sensor nodes at the leaf as shown in Fig.1 where *m* is the degree of the tree. The tree is maintained by the cluster head which is CH-node. In Fig.1. s_0,s_2, …. , s_8 represent sensor nodes within a cluster. Nodes within a cluster are again organized into smaller groups (called as *subgroups*) of fixed size based on the *m* value. This type of grouping reduces rekey operation when a node is compromised. Every sensor node shares a key with the cluster head called its private key used to communicate with the cluster head securely, nodes k_0,k_1,…., k_8 correspond to private keys. The keys k_{0-2}, k_{3-5}, k_{6-8} represent the keys that are shared by some subset of sensors (called as intermediate keys). Intermediate keys are used for intra group communication within a cluster. Key at the root of the tree is the cluster

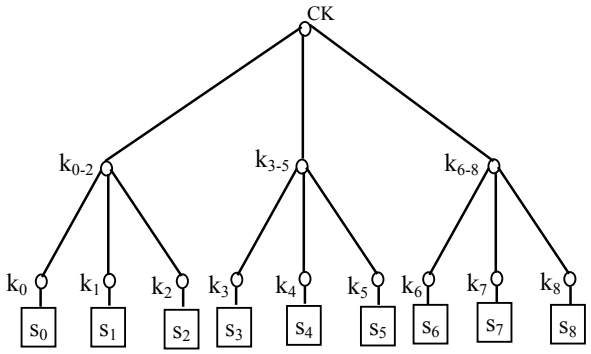

Fig.1. s_0-s_8 are sensor nodes in a cluster and k_0 to k_8 are pre loaded private keys of sensors, k_{0-2}, k_{3-5}, k_{6-8} are auxiliary keys and CK is the cluster key

key (CK). CK is shared by all the nodes in the cluster. Nodes within a cluster can communicate securely using CK. Every sensor node will store all the keys along the path from leaf to root of the tree. All CH-nodes in the network form another *m*-ary tree which is maintained by base station. We call the key that is shared by all CH-nodes as Common Cluster head Key (CCHK). CH-nodes can communicate with each other using the key CCHK.

A. Security analysis of Tree-Based Scheme

Security analysis is explained in this section by considering the following issues : Key establishment,

DOI: 01.ijns.01.01.05

rekey operation as a result of events like node addition, node compromise and key refresh at regular intervals.

Key establishment : Each sensor is pre loaded with a private key that it shares with its cluster head before deployment. Initially all CH-sensors are pre loaded with all the keys that are assigned to sensor nodes. After deployment all CH-sensors broadcast hello message to SN-sensors. Each SN-sensor selects the nearest CH-sensor as its cluster head. After receiving reply from SN-sensors each CH-sensor will delete the keys of SN-sensors that are not there in its cluster. Each CH-sensor will now construct a *m*-ary tree and assigns keys for each node in the tree as explained in section III. Now, initially CH-sensor will distribute all the keys along the path from leaf to root of respective nodes by encrypting the keys using private keys of the sensors. Upon receiving the set of keys, SN-sensors can communicate with cluster head as well as other sensors with in the cluster using the cluster key CK.

Node Revocation / Node compromise : We assume that we have intrusion detection mechanism to detect node compromise. As soon as a node is compromised corresponding cluster head will change all the keys that are known to compromised node (i.e., keys along the path from compromised node's position to root of the tree). The changed keys are distributed securely to existing nodes. For e.g. if say node s_4 is compromised, keys k_{3-5} and CK are changed to k'_{3-5} and CK'. First, k'_{3-5} is encrypted using k_3 and k_5 and CK' is encrypted using k_{0-2}, k'_{3-5}, k_{6-8}. Nodes s_3 and s_5 can decrypt the new intermediate key k'_{3-5} using the keys k_3 and k_5. Now, nodes s_0,s_1,s_2 can decrypt the new cluster key CK' using the key k_{0-2}, s_3 and s_5 decrypt using k'_{3-5} and nodes s_6,s_7,s_8 can decrypt using the key k_{6-8}. If a single node is compromised the number of encryptions required to distribute new set of keys securely is *m*(h-1) where h is the height of the tree.

Addition of New node : A new node is pre loaded with a private key that it shares with the cluster head. Base station encrypts the private key of the new SN-sensor using the CCHK that is maintained for cluster heads and the same is sent. Upon receiving the message from base station each CH-sensor will have the information regarding the new node. Each CH-node will now broadcast Hello message to newly added SN-sensor. Now as in initial setup phase SN-sensor will choose nearest CH-sensor as its cluster head. Now the cluster head will find an appropriate position for the new node in the tree and tree is updated (i.e., all the keys along the path including the cluster key are changed). Cluster head will now distribute new set of keys to corresponding nodes as well as the new node will receive all the keys along the path. In order to distribute the changed keys securely cluster head uses private key of the new node and for other nodes it uses previous cluster key CK.

Key Refresh : In order to achieve key freshness it is required to change the cluster key CK as well as Common Cluster Head Key CCHK periodically. The cluster key CK is changed to CK' by respective cluster heads and is distributed securely to nodes in the cluster by encrypting the CK' using old cluster key CK. Similarly base station will change CCHK to CCHK' and distributes it to all CH-nodes securely by encrypting CCHK' using CCHK.

IV. Description of the CRT-based scheme

In this section we explain the basic Chinese remainder theorem, followed by the detailed description of the protocol for key establishment using CRT.

A. Chinese Remainder Theorem

Let the numbers $m=m_1,m_2,\ldots,m_t$ be positive integers which are prime in pair, i.e., $\gcd(m_i,m_j)=1$ for $i \neq j$. Furthermore, let $b_1,b_2, \ldots, b_t$ be integers. Then the system of congruences defined below has a simultaneous solution X to all of the congruences and any two solutions are congruent to one another modulo m.

$$x \equiv b_1 \pmod{m_1}$$
$$x \equiv b_2 \pmod{m_2}$$
$$\vdots$$
$$x \equiv b_t \pmod{m_t}$$

The solution for this congruence system is obtained by solving $X=\sum_{i=1}^{t} b_i C_i$. Where $c_i=M_i(M_i^{-1} \mod m_i)$ and $M=\prod_{i=1}^{t} m_i$, $M_i = M/ m_i$. M_i^{-1} is multiplicative inverse of M_i, to find multiplicative inverse Extended Euclid's Algorithm is used.

B. Key Establishment using CRT-based Scheme

Following steps explain the key establishment process.
Initialization : Every SN-sensor is loaded with private key k_i that it shares with the base station. The key k_i is chosen from a key pool K of pairwise relatively prime numbers. Initially each CH-sensor is loaded with the information of all SN-sensors in the network (i,e,Node ID and its corresponding private key of all the nodes are stored).

Cluster Formation : After deployment all CH's (i.e., CH-sensors) broadcast Hello message to other sensors in the network. Each sensor selects the nearest CH-sensor as its Cluster Head. After receiving reply from SN-sensors each Cluster Head will delete the keys of the SN-sensors that are not there in its cluster.
Building Congruence System : In this step each CH selects initial cluster key CK and constructs congruence system using this initial key as follows :

$$X \equiv p_1 \pmod{k_1}$$
$$X \equiv p_2 \pmod{k_2}$$
$$\vdots$$
$$X \equiv p_n \pmod{k_n}$$

Where $p_1 \leftarrow CK \oplus k_1$, $p_2 \leftarrow CK \oplus k_2 \ldots p_n \leftarrow CK \oplus k_n$ and $k_1 \ldots k_n$ are private keys assigned to each node

DOI: 01.ijns.01.01.05

which are chosen from a pool of pairwise relatively prime numbers.

Find Solution : Cluster Head will solve the congruence system and compute the value of X.

Broadcast X: The X value computed by solving the congruence system in the previous step is broadcasted such that other nodes in the network will receive this value.

Key computation by other nodes : Each sensor will now compute the CK as : $CK \leftarrow((X \bmod k_i) \oplus k_i)$.

C. Security analysis of CRT-Based Scheme

In this section we explain in detail the events like node addition, node compromise and key refresh at regular interval.

Node Revocation / Node compromise: We assume that we have intrusion detection mechanism to detect node compromise. As soon as a node is compromised corresponding cluster head will construct new congruence system as explained in the *Building congruence system* phase of section IV B. Suppose if the compromised node is s_j with private key k_j then the new congruence system constructed by Cluster Head is

$$\begin{aligned} X &\equiv p_1 \pmod{k_1} \\ X &\equiv p_2 \pmod{k_2} \\ &\vdots \\ X &\equiv p_{j-1} \pmod{k_{j-1}} \\ X &\equiv p_{j+1} \pmod{k_{j+1}} \\ &\vdots \\ X &\equiv p_n \pmod{k_n} \end{aligned}$$

For the above congruence system cluster head will find the solution X and broadcast the X value. Now other nodes in the cluster except the compromised node will be able to compute the new cluster key CK' using the X value as explained in above protocol.

Addition of New node : A new node is pre loaded with a private key that it shares with the cluster head. Base station encrypts the private key of the new SN-sensor using the CCHK that is maintained for cluster heads and the same is sent. Upon receiving the message from base station each CH-sensor will have the information regarding the new node. Each CH-node will now broadcast Hello message to newly added SN-sensor. Now as in initial setup phase SN-sensor will choose its cluster head. After the node is admitted to a particular cluster, in order to compute new key CK' the cluster head will build new congruence system, if the new node added is say s_j with private key k_j then the congruence system constructed by cluster head is :

$$\begin{aligned} X &\equiv p_1 \pmod{k_1} \\ X &\equiv p_2 \pmod{k_2} \\ &\vdots \\ X &\equiv p_j \pmod{k_j} \\ &\vdots \\ X &\equiv p_n \pmod{k_n} \end{aligned}$$

The cluster head solve the above congruence system and broadcast the computed X value. Upon receiving the X value each node will compute the required cluster key CK'.

Key Refresh : In order to achieve key freshness it is required to change the cluster key CK periodically. To change the key at regular interval the Cluster Head will choose a new CK' and for the selected CK' it builds new congruence system and solve the system to compute new X value. CH broadcasts the computed X value so that other nodes can compute the new CK'.

V. PERFORMANCE ANALYSIS

Storage : In the *Tree-Based Scheme* each SN-sensor is required to store $\log_m n$ keys (i.e., keys along the path from leaf to root of the tree) where n is the number of nodes in a cluster and *m* degree of the tree. Each CH-sensor is required to store $\sum_{i-0}^{h} mi$ keys, where h is the height of the tree and *m* the degree of the tree. In the *CRT-Based Scheme* no additional storage is required by SN-sensors, each SN-sensor will store only its private key k_i. For the scheme in [8] the storage is : for a key sharing probability of 0.8 SN-sensor stores 5 generation keys and CH-sensor approximately 250 generation keys.

Computation : In *Tree-Based Scheme*, computation costs are measured in terms of number of encryptions. Total number of encryptions performed by cluster head (CH-node) in case of node addition are 2(h-1) where h is the tree height. For node addition computation with respect to SN-sensor not in the path of the joining node is one and for the SN-sensor in the path of joining node computation is equal to (h-1) decryptions. When a single node is compromised, total number of encryptions are m(h-1).

In order to compute the new cluster key CK' in *CRT-Based Scheme* each SN-sensor in the cluster is required to perform one modulus operation and one EXOR operation. CH-sensor is required to solve the congruence system as a result of events like node addition, node compromise or to refresh the key at regular interval. The computation cost incurred at cluster head to solve the congruence system is $O(t(\log m)^3) + O(t(\log m)^2)$ as per the analysis of Chinese Remainder Theorem in [12].

Communication : Communication cost are studied in terms of number of messages that are exchanged in order to change the required keys. In Tree-Based Scheme for events like node addition and node compromise, the number of messages constructed and communicated vary from one to $\log_m n$ which is the communication cost incurred at CH-sensor. Similarly each SN-sensor performs either one or $\log_m n$ receive operations. For key refresh each CH-sensor performs one transmit operations and SN-sensor one receive operation in order to update the cluster key.

In *CRT-Based Scheme*, when a node is added or compromised the cluster head constructs new congruence system in order to change the key. The computed X value is distributed using single broadcast message to other nodes in the cluster. Each SN-sensor performs one receive operation to get the value of X, using which they can compute the key CK'. The communication cost incurred in the *CRT-Based Scheme* is : one transmit operation by CH-sensors(cluster head) and one receive

DOI: 01.ijns.01.01.05

operation by SN-sensors(other nodes in the cluster). The table below summarizes the communication costs incurred by the proposed schemes and the scheme in [8].

TABLE 1: DEPICTS STORAGE AND COMMUNICATION REQUIREMENT FOR THE PROPOSED SCHEMES AND THE SCHEME BY SAJID ET.AL.

	Storage		Communication	
	CH Sensor	SN Sensor	CH Sensor	SN Sensor
Tree Based Scheme	$\sum_{i-0}^{h} mi$	$\log_m n$	1 to $\log_m n$ transmit	1 to $\log_m n$ receive
CRT Based Scheme	n+1	2	1 Broadcast Message	1 receive Operation
Scheme By Sajid et.al	250(appx) keys	5(appx) keys	n transmit + n receive	2 transmit + (p+1) receive

VI. CONCLUSION

The paper presents new schemes for key management for confidential communication between node and its cluster head in hierarchical sensor networks. The schemes are analyzed in detail with respect to security and performance. Performance analysis shows that *Tree-Based Scheme* exhibits better performance which achieves rekey operation by performing $\log_m n$ communications with some additional storage. In *CRT-Based Scheme* key is established in an efficient way for node addition, node compromise and also at regular intervals. The communication cost incurred at each node for establishing key is one receive operation and computation cost incurred is one modulus operation and one EX-OR operation by each node.

REFERENCES

[1] H.Chan and A.Perrig. and D.Song. Random key pre distribution schemes for sensor networks. *IEEE symposium on Research in Security and Privacy*, pages 197-213, 2003.

[2] Y.Cheng and D.P.Agrawal. Efficient pairwise key establishment and management in static wireless sensor networks. *In Second IEEE International Conference on Mobile ad hoc and Sensor Systems*, 2005.

[3] W.Du, J.Deng, Y.S.Han, and P.K.Varshney. A pairwise key pre distribution scheme for wireless sensor networks. *In Proc. of the 10th ACM conference of Computers and Communication Security (CCS'03)*. pp.42-51, 2003.

[4] X.Du, M.GUizani, Y.Xiao, S.Ci, and H.H.Chen, A Routing-Driven Elliptic Curve Cryptography based Key Mangement scheme for Heterogeneous Sensor Networks. *IEEE transactions on Wireless Communications*.

[5] X.Du and F.Lin. Maintaining Differential coverage in heterogeneous sensor network. *EURASIP Journal of Wireless Communications and Networking*, (4):565-572, 2005.

[6] L.Eschenauer and V.D.Gligor. A key management scheme for distributed sensor networks. *In Proceedings of the 9th ACM conference Computer and Communications security*, pages 41-47, November 2002.

[7] J.Hwang and Y.Kim. Revisiting random key pre distribution schemes for WSN. *In Proc. of the 2nd ACM workshop on Security of ad hoc and sensor networks* , pp.43-52, 2004.

[8] S.Hussain, F.Kausar, and A.Masood. An Efficient Key Distribution Scheme for Heterogeneous Sensor Networks. *IWCMC'07*, 2007.

[9] D.Liu and P.Ning. Establishing pairwise keys in distributed sensor networks. *In proceedings of the 10th ACM conference on Computers and Communication Security (CCS'03)*. pp.52-61, 2003.

[10] K.Lu, Y.Qian and J.Hu. A framework for distributed key management schemes in heterogeneous wireless sensor networks. *In IEEE International Performance Computing and Communications Conference*, pages 513-519, 2006.

[11] R.D.Pietro, L.V.Mancini, and A.Mei. Random Key assignment to secure wireless sensor networks. *In 1st ACM workshop on Security of Ad Hoc and Sensor Networks*, 2003.

[12] Samuel S. Cryptanalysis of Number Theoretic Ciphers. *CRC Press*, 2003.

[13] C.Wong, M.Gouda, and S. Lam, Secure Group Communication Using key Graphs. *In proceedings of the ACM SIGCOMM'98*, Oct.1998.

[14] Xinliang Zheng, Chin-Tser Huang and Manton Matthews, Chinese Remainder Theorrem Based Group Key Management. *ACMSE*, 2007.

[15] S.Zhu, S.Setia, and S.Jajodia. LEAP : Efficient Security Mechanisms for Large Scale Distributed Sensor Networks. *In Proc. of 10th ACM Conference on Computers and Communication Security (CCS'03)*, 2003.

DOI: 01.ijns.01.01.05